ON EXPLOITING HUMAN DOMAIN WORKFLOWS IN CYBER-PHYSICAL SYSTEMS

BY

HONGWEI WANG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2016

Urbana, Illinois

Adviser:

Professor Tarek Abdelzaher

ABSTRACT

In this thesis, we describe a general methodology for enhancing sensing accuracy in cyber-physical systems that involve human domain workflows in noisy physical environment. A novel *workflow-aware sensing model* is proposed to jointly correct unreliable sensor data and keep track of states in a workflow. We also propose a new inference algorithm to handle cases with partially known states and objects as supervision. Our model is evaluated with extensive simulations. As a concrete application, we develop a novel log service called *Emergency Transcriber*, which can automatically document operational procedures followed by teams of first responders in emergency response scenarios. Evaluation shows that our system has significant improvement over commercial off-the-shelf (COTS) sensors and keeps track of workflow states with high accuracy in noisy physical environment.

To my parents, for their love and support.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my adviser Professor Tarek Abdelzaher for his patience, support and inspiration on this work. His guidance helped me in all the time of research and writing of this thesis.

I also would like to appreciate my fellow labmates for their help in this project. In particular, Yunlong Gao and Shaohan Hu made great effort on problem definition and early contribution to this work. Shiguang Wang provided me with insightful suggestions and encouragement. This thesis cannot be accomplished without their support.

Lastly, I would like to thank my parents for their love, encouragement and support on every aspect of my life.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Tasks executed by teams of first responders are often critical and risky, where failures may cause serious damage or even loss of life. Examples include medical emergency [1], firefighting [2] and disaster response [3]. Since it is often stressful to handle these tasks, human teams must follow well-established workflows to reduce risk and improve efficiency.

In these critical tasks, a log service is often required to keep track of the operations that human teams perform as well as record the parameters and outputs at each stage. It is useful for (i) early detection of procedure mistakes, (ii) log of events for future reference, and (iii) better collaboration among team members, providing a consistent joint understanding of execution steps in the procedure workflow.

Traditionally, the procedure steps are manually logged by human, which is labor intensive. Nowadays, there is an increasing popularity in deploying cyber-physical systems with a set of sensors to log and monitor states and parameters [4, 5]. However, since critical tasks are often performed in noisy or extreme physical environment, commercial off-the-shelf (COTS) sensors may have low accuracy when directly deployed, especially for those with complex outputs, such as voice recognition and computer vision. On the other hand, it can be expensive to get custom-made sensors adaptive to the new environment. In this thesis, we propose a general approach, by treating the COTS sensors as a blackbox, and correct the sensor outputs in a postprocessing manner by considering physical constraints and situation awareness according to the workflow followed by human teams. Besides, our approach can infer the states of the workflow that human teams have operated, offering a high level states tracking service.

In this thesis, we assume that human interactions with cyber-physical systems evolve according to a predefined workflow. The workflow can be obtained, for example, from an operations manual. Each state of the workflow

is associated with actions that team members are allowed to perform. These actions have sensory signatures. Hence, a different expectation for sensor values exist in different states. It therefore becomes possible to use the sequence of received sensor measurements to jointly estimate both (i) the state transitions experienced by individuals following the workflow, and (ii) the most likely measured values given the obtained noisy measurements and the expected state-specific ground-truth value distribution, i.e. correcting the unreliable sensor outputs. In this thesis, we focus on discrete sensor outputs. We show how this problem can be formulated with a novel *workflow-aware sensing model* and evaluate its effectiveness on improving the accuracy of raw sensor measurements.

We first evaluate the performance of workflow-aware sensing model through simulations, where abstract workflow states are associated with sets of possible measured values in the physical world, and a noisy sensor with unreliable outputs is simulated to emit values. As a concrete application of our model, we develop a novel log service for teams of first responders, called *emergency transcriber*. It constitutes an audio interface for reliably recording and disseminating situation progress as extracted from the team's audio communications. As noted above, such teams typically follow predefined collaborative workflow as dictated by the relevant engagement protocols, specifying their roles and communications. Given the critical nature of the situation, the vocabulary used is often constrained and dependent on the current stage of the workflow being executed. The emergency transcriber documents the sequence of procedure steps executed by the team as well as their parameters, if any (e.g. dosage of medications administered). As a case study, we conduct a physical experiment involving a medical scenario based on the adult cardiac arrest workflow. Our evaluation demonstrates that we are able to achieve 80% accuracy in workflow state identification and when relying on a COTS sensor of only 40% accuracy in noisy voice recognition. When the accuracy of the underlying acoustic sensor grows to 77%, our state estimation is close to 100% correct.

This thesis covers and greatly extends our previous work [6]. The main contributions are listed as follows:

- We exploit structured human interactions (i.e. workflows) to enhance sensing accuracy and keep track of states in cyber-physical systems. A novel workflow-based sensing model is proposed to solve the problem.

- We extend the basic model with a new inference algorithm to handle cases with partially known states and/or objects as supervision.
- Our model is evaluated with extensive simulations, which shows its effectiveness in different conditions based on 100,000 randomly generated workflows.
- We have developed a novel log service for human teams of first responders to keep track of executed workflow states and recognize communication keywords.

The rest of the thesis is organized as follows. We formulate our problem in Chapter 2 and introduce our workflow-aware sensing model in Chapter 3. Some practical issues of applying this model are presented in Chapter 4. Our model is evaluated with extensive simulations in Chapter 5. A case study evaluation with a novel application of emergency transcriber in medical environment is presented in Chapter 6. Related work is covered in Chapter 7. The thesis concludes in Chapter 8.

CHAPTER 2

PROBLEM FORMULATION

Fig. 2.1 shows a simple abstract workflow topology of an emergency procedure. The nodes represent states (or stages) of the procedure in the workflow. We assume state transitions take place as a Markov chain. The number on the edge indicates the probability of state transitions. Each state is associated with a probability of emitting certain ground-truth data objects. For example, in a medical workflow of inspecting a person's airway, physicians may utter words such as "air", "breath", "lung", "airway", "obstructed", "clear" and so on. These words correspond to the ground truth objects emitted in the aforementioned state. They are recorded and recognized by a voice recognition sensor in cyber-physical systems. However, since the physical environment is noisy, the performance of the COTS sensor may not be reliable. For example, it may recognize "lung" as "long" by mistake. Our goal is to correct the sensor data as well as infer the sequence of states (i.e. steps that human teams have executed) given the noisy values emitted from the COTS sensors. In this paper, we focus on discrete sensor values, and cast the challenge as a classification problem.

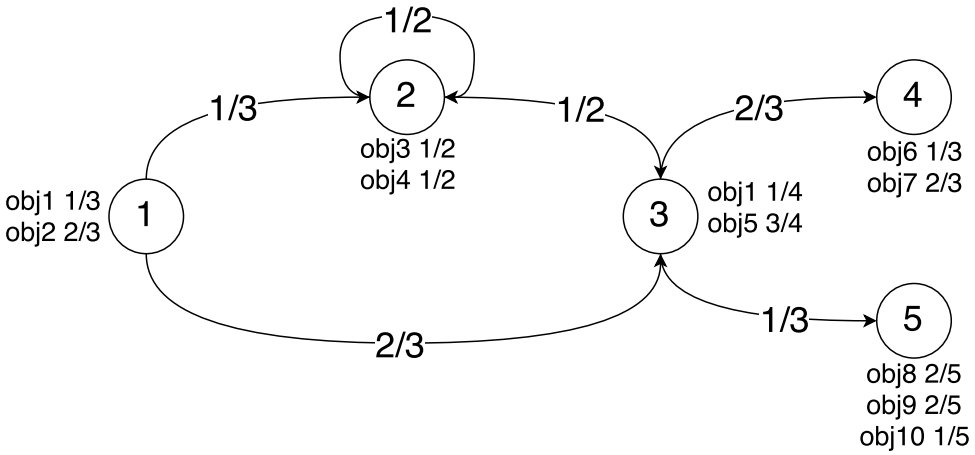


Figure 2.1: An example workflow

Formally, we define the workflow as a directed graph with a set of states S that follows a states transition probability matrix T, where $T_{i,j}$ indicates the probability of state S_i to state S_j. Each state is associated with a distribution of emitted objects O. The objects emission distribution for each state is denoted as E, where $E_{i,j}$ indicates the probability of object O_j emitted from state S_i. We also define a confusion matrix C, where $C_{i,j}$ is the probability that object O_i is recognized as O_j by the COTS sensor.

In practice, based on different conditions, not all states are covered in one execution of critical tasks. We define a path as a sequence of states that are actually executed by human teams, denoted as $\mathbf{z} = (z_1, z_2, ..., z_N)$ for the time $t = 1, ..., N$, where each z_i is chosen from the state space S according to state transition probability matrix T. We use I to denote the initial states probability distribution at time $t = 1$. An object is emitted from each state as ground-truth value, i.e. $\mathbf{x} = (x_1, x_2, ..., x_N)$, where each x_i is chosen from the object space O following the objects emission probability matrix E. A COTS sensor will recognize the objects as outputs $\mathbf{y} = (y_1, y_2, ..., y_N)$, where each y_i is also chosen from the object space O, by following the confusion matrix C. Table 2.1 shows a summary of notations.

The parameters of the model can be obtained from domain knowledge or learned from historical data. For example, the state transition probability can be calculated as

$$T_{ij} = \frac{count(z_k = S_i \wedge z_{k+1} = S_j)}{count(z_k = S_i)}$$

Intuitively, it means the probability of state S_i transiting to state S_j equals with number of times that state S_j is the next state of state S_i divided by number of times that state S_i appears. Similarly, object omission probability can be calculated as

$$E_{im} = \frac{count(z_k = S_i \wedge x_k = O_m)}{count(z_k = S_i)}$$

Intuitively, it means the probability of object O_m emitted from state S_i equals with number of times that object O_m emitted from state S_i divided by number of times that state S_i appears. Sensor confusion probability may be difficult to learn from the sparse raw data, which can be approximated with sensor accuracy or objects similarities.

S	States space
O	Objects space
I	Initial states probability distribution
T	States transition probability matrix
E	Objects emission probabilities at each state
C	Sensor objects confusion probability matrix
$\mathbf{x}$	Variables of ground-truth objects
$\mathbf{y}$	Variables of raw sensor outputs
$\mathbf{z}$	Variables of states sequence

Table 2.1: Summary of notations

As noted above, only raw sensor outputs are observed while sequence of states and actual objects are hidden. Our goal is to find the sequence of states $\mathbf{z}$ and the sequence of objects $\mathbf{x}$ that maximize the posterior probability $p(\mathbf{zx}|\mathbf{y})$, based on inaccurate measurement of $\mathbf{y}$. Mathematically, we write this as follows:

$$\mathbf{zx} = \arg\max_{\mathbf{zx}} p(\mathbf{zx}|\mathbf{y})$$

CHAPTER 3

WORKFLOW-AWARE SENSING MODEL

Our workflow-aware sensing model is motivated by Hidden Markov Model (HMM) [7], which is widely used in sequence labeling tasks. In HMM, the states are hidden and the objects emitted at each state are observed. The goal of HMM is to infer the sequence of states based on sequence of objects observed. Compared with HMM, our task is more complicated. We only observe the sensor outputs, while the actual emitted objects and states are hidden, and our goal is to infer both of them. The general idea is to exploit workflow information (state transition and object emission matrix) and sensor information (confusion matrix) as constraints to achieve optimal solution.

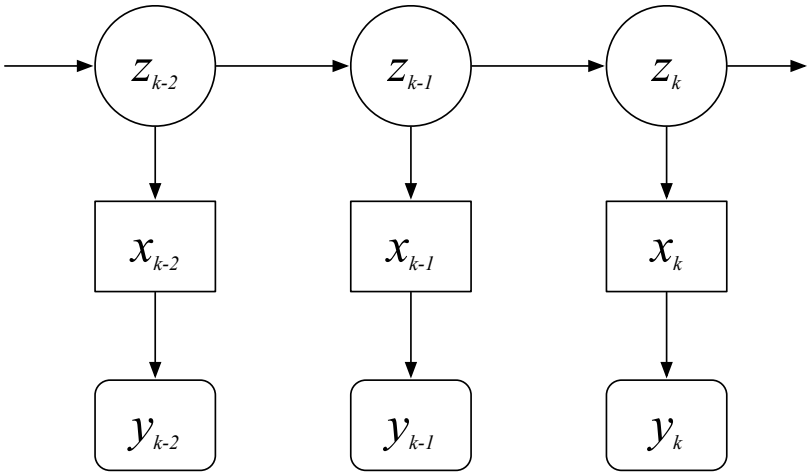


Figure 3.1: Workflow-aware sensing model

Our workflow-aware sensing model is shown in Fig. 3.1. The states sequence $\mathbf{z} = \{z_1, z_2, ..., z_n\}$ is generated by following initial states probability distribution I and states transition probability matrix T, i.e.

$$p(z_1 = S_i) = I_i, i = 1...|S|$$

$$p(z_k = S_j | z_{k-1} = S_i) = T_{i,j}, i, j = 1...|S|, k = 2...N$$

At each state, an object is emitted according to objects emission probability matrix E, i.e.

$$p(x_k = O_m | z_k = S_i) = E_{i,m}, i = 1...|S|, m = 1...|O|, k = 1...N$$

For each object, the sensor will generate a corresponding output according to the confusion matrix C, i.e.

$$p(y_k = O_n | x_k = O_m) = C_{m,n}, m, n = 1...|O|, k = 1...N$$

According to Bayes' theorem and conditional independence in our model, we would like to infer the most likely states and objects sequence $\mathbf{zx}$ based on the observed objects sequence $\mathbf{y}$ i.e.

$$\begin{aligned}
\mathbf{zx} &= \arg\max_{\mathbf{zx}} p(\mathbf{zx}|\mathbf{y}) \\
&= \arg\max_{\mathbf{zx}} p(\mathbf{zxy}) \\
&= \arg\max_{\mathbf{zx}} [p(z_1) \prod_{k=2}^{N} p(z_k|z_{k-1}) \prod_{k=1}^{N} p(x_k|z_k) \prod_{k=1}^{N} p(y_k|x_k)]
\end{aligned}$$

Solving the above equation by exhaustively listing all possible states and objects sequence will require $O((|S| \times |O|)^N)$ operations. Instead, we propose a dynamic programming algorithm as shown in Algorithm 1. The algorithm takes the observed objects sequence $\mathbf{y}$, states space S, objects space O, initial states distribution I, states transition matrix T, objects emission matrix E, and sensor confusion matrix C as input. The output of the algorithm is the optimal sequence of states $\mathbf{z}$ and objects $\mathbf{x}$. We use $MP_{k,i}$ to denote the maximum joint probability of reaching state i at sequence k. $MS_{k,i}$ denotes the previous state that transits to state i at sequence k to achieve $MP_{k,i}$. $MO_{k,i}$ denotes the object emitted in state i at sequence k to achieve $MP_{k,i}$.

First, we initialize $MP_{1,i}, MS_{1,i}, MO_{1,i}$ in Line 1-5. $MP_{1,i}$ equals with initial probability I_i times the maximum value of $E_{i,m} * C_{m,y_1}$ by comparing all possible objects $m = 1...|O|$. $MO_{1,i}$ equals with the object m which achieves $MP_{1,i}$. Since it is the first state, we can simply set $MS_{1,i}$ to be 0.

Line 6-10 shows the recursion of our algorithm, with the core formula in line 8. To calculate $MP_{k,i}$, we need to consider all previous states $j = 1...|S|$ that

Algorithm 1: Workflow-aware Sensing Model Inference

WSM-INFER$(S, O, I, T, E, C, \mathbf{y})$ return $(\mathbf{z}, \mathbf{x})$

1: **for** $i \leftarrow 1...|S|$ **do**
2: $\quad MP_{1,i} \leftarrow I_i * \max_{m=1...|O|}\{E_{i,m} * C_{m,y_1}\}$
3: $\quad MS_{1,i} \leftarrow 0$
4: $\quad MO_{1,i} \leftarrow m$ which achieves $MP_{1,i}$.
5: **end for**
6: **for** $k \leftarrow 2...N$ **do**
7: $\quad$ **for** $i \leftarrow 1...|S|$ **do**
8: $\quad\quad MP_{k,i} \leftarrow \max_{j=1...|S|,m=1...|O|}\{MP_{k-1,j} * T_{j,i} * E_{i,m} * C_{m,y_k}\}$
9: $\quad\quad MS_{k,i} \leftarrow j$ which achieves $MP_{k,i}$
10: $\quad\quad MO_{k,i} \leftarrow m$ which achieves $MP_{k,i}$
11: $\quad$ **end for**
12: **end for**
13: $z_N \leftarrow \arg\max_i MP_{N,i}$
14: $x_N \leftarrow MO_{N,z_N}$
15: **for** $k \leftarrow N-1...1$ **do**
16: $\quad z_k \leftarrow MS_{k+1,z_{k+1}}$
17: $\quad x_k \leftarrow MO_{k,z_k}$
18: **end for**

can transit to state i at sequence k as well as all possible objects $m = 1...|O|$ emitted at state i and use the maximum value of $MP_{k-1,j} * T_{j,i} * E_{i,m} * C_{m,y_k}$ as $MP_{k,i}$.

Lastly from line 13 - 18, we show how **zx** can be derived with the auxiliary variables MP, MS, MO calculated above. **zx** is derived in reverse order from N to 1. We calculate the last state first. $MP_{N,i}$ stores the maximum joint probability of reaching state i at sequence N. To find most likely z_N, we only need to compare all $MP_{N,i}, i = 1...|S|$ and assign z_N to the i which achieves the maximum value. With last state z_N, last actual object x_N is MO_{N,z_N}. Next, we can derive z_k and x_k in reverse order, where z_k is $MS_{k+1,z_{k+1}}$, as $MS_{k+1,z_{k+1}}$ stores the the previous state z_k that transits to state z_{k+1} at sequence $k+1$ to achieve the maximum probability $MP_{k+1,z_{k+1}}$. x_k is just MO_{k,z_k}.

The complexity of Algorithm 1 is $O(N \times |S|^2 \times |O|)$, where N denotes number of sequence, $|S|$ denotes number of states, and $|O|$ denotes number of objects. It is significant lower than exhaustive search which takes $O((|S| \times |O|)^N)$ time.

CHAPTER 4

PRACTICAL ISSUES

In this section, we discuss some practical issues to the workflow-aware sensing model when applying it to real application scenarios.

4.1 Partially Known States and Objects as Supervision

In practice, people may have already known or recorded some states and objects, such as initial state/object, last state/object, or any states/objects along the path, providing a supervision to our system, which can be utilized to guide the states and objects inference. We modify Algorithm 1 to adapt to the changes, as shown in Algorithm 2. We use KS to denote the input of known states, and KO to denote the input of known objects. $KS_k = i$ means state is S_i at sequence k, while $KS_k = 0$ means state is unknown at sequence k. Similarly, $KO_k = m$ means object is O_m at sequence k, while $KO_k = 0$ means object is unknown at sequence k.

If the first state is known (KS_1), the initial states probabilities can be updated for this particular inference by setting I_{KS_1} to be 1 and others to be 0, as shown in line 1 - 6. If the first object is known (KO_1), we can simply ignore other objects, i.e. $MO_{1,i} = KO_1$ and $MP_{1,i} = I_i * E_{i,KO_1} * C_{KO_1,y_1}$, as shown in line 8 - 10. If a state at sequence k is known (KS_k), the probability of reaching any state i except KS_k at sequence k should be 0, i.e. $MP_{k,i} \leftarrow 0$, shown in line 19 - 20. If an object at sequence k is known (KO_k), then we can simply ignore other objects, i.e. $MO_{k,i} = KO_k$ and $MP_{k,i} \leftarrow \max_{j=1...|S|}\{MP_{k-1,j} * T_{j,i} * E_{i,KO_k} * C_{KO_k,y_k}\}$, as shown in line 21 - 24. The other parts of the algorithm remains the same. The time complexity of Algorithm 2 is $O(N \times |S|^2 \times |O|)$.

4.2 Smoothing Model Parameters

Knowledge of workflow only helps if human teams follow it. However, in real application, there may be some cases where human teams perform slightly different from the original workflow, such as skipping a step. Besides, the COTS sensor may miss some measurements as well. Take Figure 2.1 as an example, in which state 1 transits to state 3 with probability 2/3, but state 1 cannot transit to state 4 directly. Suppose we know that the previous state is state 1, and object 1 is emitted and recognized. Suppose the next state is state 3, and an object is emitted (e.g. object 5) but missed by the sensor. The workflow then reaches state 4, where one of the objects, say object 6, is emitted and classified correctly by the sensor. Therefore, the overall output from sensor is: object 1 followed by object 6; implying that state 1 transits to state 4 directly, which is impossible according to the predefined workflow. If we use the basic algorithm alone, it will consider the measurement of object 6 to be an error and try to match it to objects in state 3 according to the confusion matrix, thereby giving an erroneous classification result.

To make our system more robust towards workflow deviation and missing measurements, we adopt Laplace smooth to the parameters of the model such as state transition matrix to avoid 0 probability. The probability of state S_i transiting to state S_j can be calculated as

$$T_{ij} = \frac{count(z_k = S_i \wedge z_{k+1} = S_j) + 1}{count(z_k = S_i) + |S|}$$

T_{ij} is smoothed between $\frac{1}{|S|}$ (uniform distribution) and $\frac{count(z_k=S_i \wedge z_{k+1}=S_j)}{count(z_k=S_i)}$ (observation learned from historical data).

Algorithm 2: Workflow-aware Sensing Model Inference with Known States and Objects

WSM-INFER$(S, O, I, T, E, C, \mathbf{y}, KS, KO)$ return $(\mathbf{z}, \mathbf{x})$

if $KS_1 > 0$ **then**
  for $i \leftarrow 1...|S|$ **do**
    $I_i \leftarrow 0$
  end for
  $I_{KS_1} \leftarrow 1$
end if
for $i \leftarrow 1...|S|$ **do**
  if $KO_1 > 0$ **then**
    $MP_{1,i} \leftarrow I_i * E_{i,KO_1} * C_{KO_1,y_1}$
    $MO_{1,i} \leftarrow KO_1$
  else
    $MP_{1,i} \leftarrow I_i * \max_{m=1...|O|}\{E_{i,m} * C_{m,y_1}\}$
    $MO_{1,i} \leftarrow m$ which achieves $MP_{1,i}$.
  end if
  $MS_{1,i} \leftarrow 0$
end for
for $k \leftarrow 2...N$ **do**
  for $i \leftarrow 1...|S|$ **do**
    if $KS_k > 0$ and $KS_k \neq i$ **then**
      $MP_{k,i} \leftarrow 0$
    else if $KO_k > 0$ **then**
      $MP_{k,i} \leftarrow \max_{j=1...|S|}\{MP_{k-1,j} * T_{j,i} * E_{i,KO_k} * C_{KO_k,y_k}\}$
      $MS_{k,i} \leftarrow j$ which achieves $MP_{k,i}$
      $MO_{k,i} \leftarrow KO_k$
    else
      $MP_{k,i} \leftarrow \max_{j=1...|S|,m=1...|O|}\{MP_{k-1,j} * T_{j,i} * E_{i,m} * C_{m,y_k}\}$
      $MS_{k,i} \leftarrow j$ which achieves $MP_{k,i}$
      $MO_{k,i} \leftarrow m$ which achieves $MP_{k,i}$
    end if
  end for
end for
$z_N \leftarrow \arg\max_i MP_{N,i}$
$x_N \leftarrow MO_{N,z_N}$
for $k \leftarrow N-1...1$ **do**
  $z_k \leftarrow MS_{k+1,z_{k+1}}$
  $x_k \leftarrow MO_{k,z_k}$
end for

CHAPTER 5

SIMULATION

In this section, we study the performance of our workflow-aware sensing model (WSM) through extensive simulations. The simulator is implemented in C++. Below we present our simulation settings and results.

5.1 Methodology

The workflow is abstracted as a directed graph, where each node represents a state and a directed edge means a possible transition between states. Each state is associated with a set of objects. The parameters of state transition matrix and object emission matrix are randomly generated. For each workflow, a ground truth path is randomly generated based on state transition matrix, and ground truth object for each state is randomly generated according to the object emission matrix.

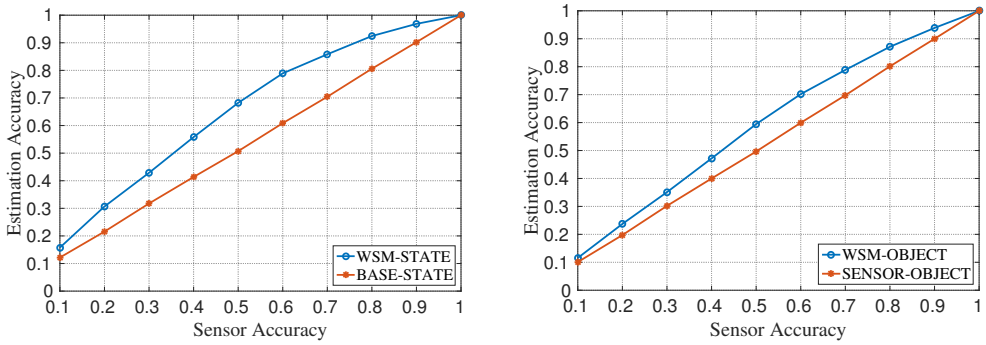
The performance of the raw sensor is simulated by setting the values in the confusion matrix parameters. We use *sensor accuracy* to capture the probability that a sensor correctly classifies a given object. For simplicity in our simulation, we assume sensor accuracy is the same for all objects, i.e. the diagonal of the confusion matrix is identical and equals with sensor accuracy. For a given object, the simulated sensor will generate its classification result based on the confusion matrix. Our model will take the sequence of sensor generated objects as inputs to infer the actual sequence of states and objects. The performance of our model is evaluated by calculating the accuracy of inferred states and objects.

The default parameters are set as follows. The workflow has 30 nodes (states) and 90 edges. Number of objects per node is 3. The ground truth path is 8 nodes length. Sensor accuracy is 0.6. We assume no states or objects along the path are known beforehand.

We use raw sensor outputs (denoted as SENSOR-OBJECT) as a baseline to evaluate the objects inference by our model (denoted as WSM-OBJECT). The baseline for state sequence inference is calculated as most likely states given raw sensor outputs based on the object emission matrix without considering state transition information of the workflow, i.e. $\mathbf{z} = \arg\max_{\mathbf{z}} P(\mathbf{y}|\mathbf{z})$. The baseline is denoted as BASE-STATE. State inference by our model is denoted as WSM-STATE. Each simulation runs for 100,000 times (i.e. 100,000 random workflows) and each result is averaged over the 100,000 executions.

5.2 Evaluation Results

First, we study how the accuracy of raw sensor affects system performance. The results are shown in Figure 5.1. More accurate sensor leads to better system performance, as expected. Our workflow-aware sensing model consistently performs better than the baselines in both state and object estimation. Note that our model does not improve too much when the sensor has either very low (e.g. 10%) or very high (more than 90%) accuracy. However, since in reality, perfect sensors rarely exist and people would normally not utilize completely unreliable sensors when building systems, our model will benefit the existing sensor systems in practice. Another observation is that our model has more improvement on state estimation than object emission. The reason is that a state can emit multiple objects (3 in our default setting), and thus objects identification is more confusing than states.

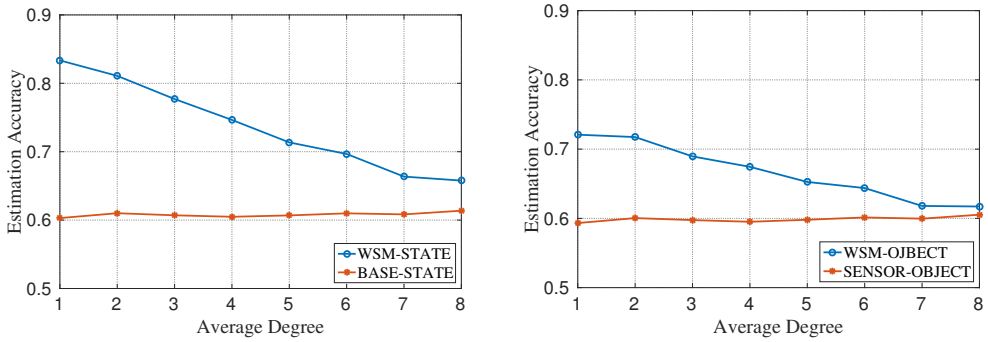


(a) State Estimation (b) Object Estimation

Figure 5.1: Performance as sensor accuracy varies.

Next, we study the system performance when the average degree of the

directed graph varies. Average degree, defined as number of edges divided by number of nodes, indicates the connectivity of the graph. From Figure 5.2, we observe that the accuracy of our workflow-aware sensing model decreases as average degree of the workflow increases. The reason is that low average degree indicates more constraints on path selection, which benefits our model on state and object estimation. On the other hand, since raw sensor and state estimation baseline do not utilize the workflow information, they remain unaffected by the average degree of the graph.



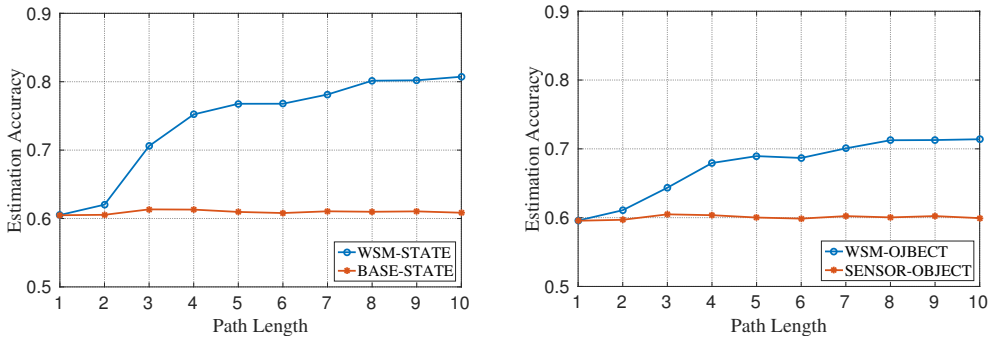
(a) State Estimation (b) Object Estimation

Figure 5.2: Performance as average degree varies.

In Fig. 5.3, we study the system performance when the path length (number of states actually executed) varies. Our workflow-aware sensing model performs better, benefiting from more context and constraints in workflow as path length increases. Since raw sensor and state estimation baseline do not utilize workflow information, the accuracy remains the same as path length varies.

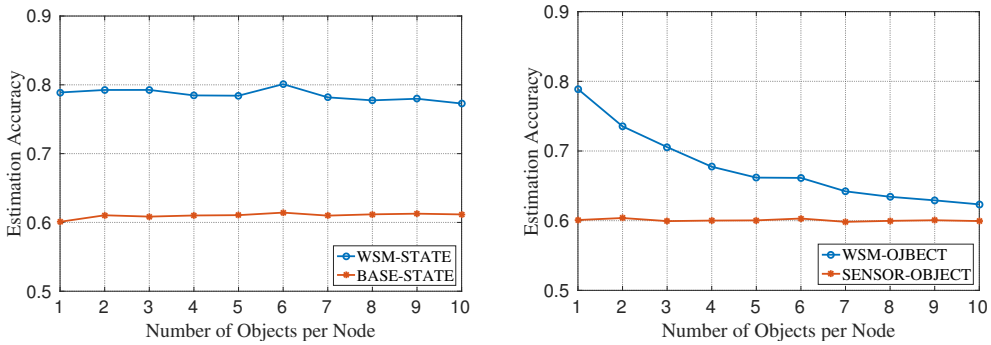
Next, we study the system performance when the number of objects per node varies. The results are shown in Fig. 5.4. We observe that state estimation of our model remains the same, which is not related with number of objects per node, while object estimation accuracy decreases when number of objects per node increases. The reason is that with more objects per state, the system is more confused to identify the correct object, but it will not affect the state estimation on the whole.

Next, we study how variations of state transition and object emission distributions in a workflow affect the performance of the system. In order to have a quantitative analysis, we assume they follow an exponential distribu-



(a) State Estimation (b) Object Estimation

Figure 5.3: Performance as path length varies.



(a) State Estimation (b) Object Estimation

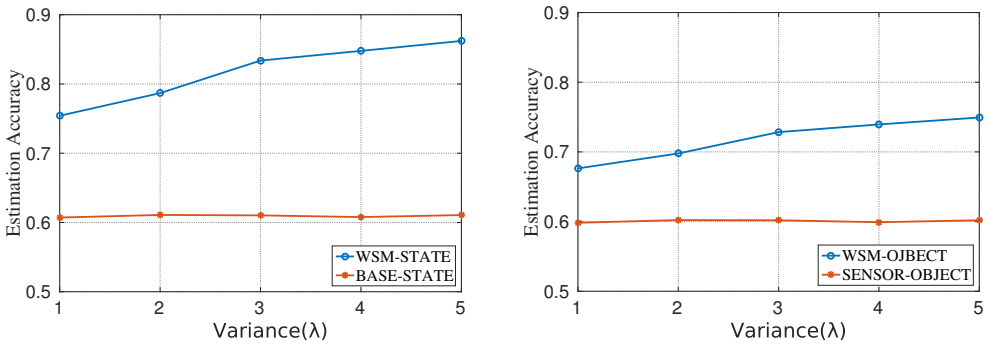
Figure 5.4: Performance as number of objects per node varies.

tion, i.e.

$$p(x) = norm(\frac{1}{\lambda^x}) = \frac{1}{\lambda^x \sum_{i=1}^{N} \frac{1}{\lambda^i}}$$

for $x = 1, 2, ..., N$, where λ is the parameter to control the variance of distributions. In state transition distribution, x is the index of possible following states given a state. In object emission distribution, x is the index of possible objects emitted at a state. For example, suppose in a workflow, state 1 is likely to transit to three states, i.e. state 2, 3, and 4. If $\lambda = 2$, $P(x = 1) = norm(\frac{1}{2}) = \frac{4}{7}, P(x = 2) = norm(\frac{1}{2^2}) = \frac{2}{7}, P(x = 3) = norm(\frac{1}{2^3}) = \frac{1}{7}$, i.e. the probability of state 1 transiting to state 2 is 2 times of state 1 transiting to state 3, and 4 times of state 1 transiting to state 4. Intuitively, a greater λ leads to a greater variance of distributions. In our experiment, we vary λ from 1 to 5. $\lambda = 1$ indicates a uniform distribution, while $\lambda = 5$ indicates a highly skewed distribution.

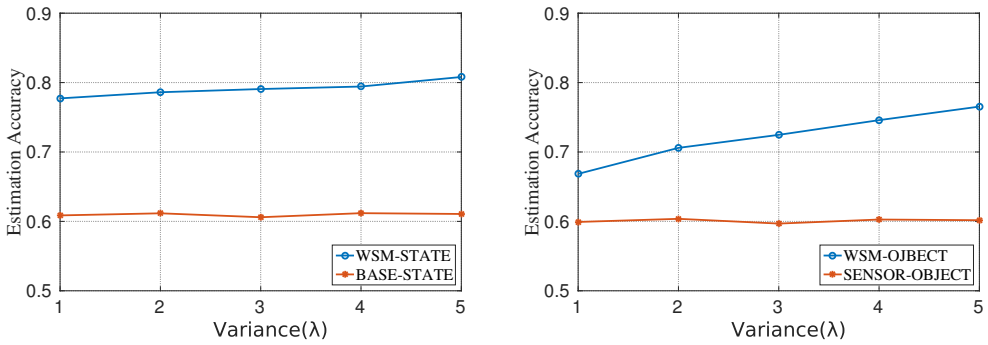
Fig. 5.5 shows the system performance when state transition distribution varies. With greater variance (i.e. greater λ), our model achieves better performance on state estimation since a state tends to be more biased towards transiting to next state. Object estimation also improves because of a better state estimation.



(a) State Estimation (b) Object Estimation

Figure 5.5: Performance as state transition distribution varies.

Fig. 5.6 shows the system performance when object emission distribution varies. With greater variance (i.e. greater λ), our model achieves better performance on object estimation since a state tends to be more biased towards emitting an object. However, since improvement on object estimation is mainly due to biased object emission within a state instead of cross states, state estimation does not improve too much.

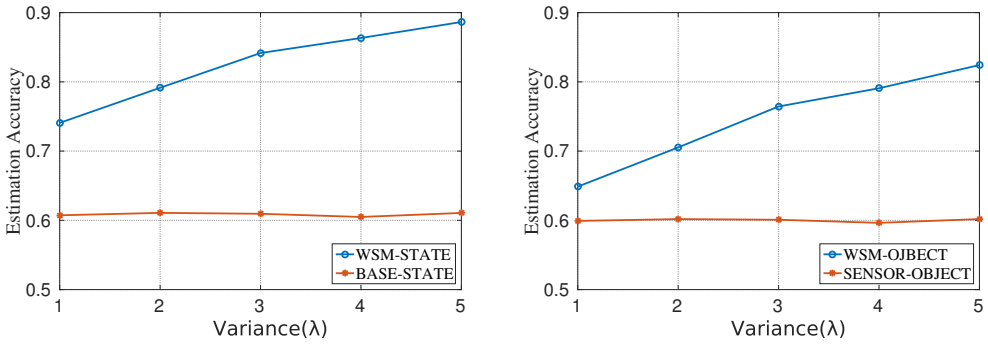


(a) State Estimation (b) Object Estimation

Figure 5.6: Performance as object emission distribution varies.

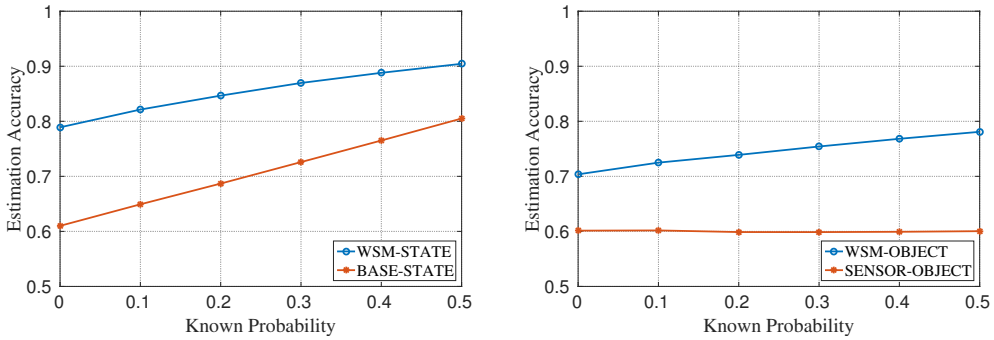
Fig. 5.7 shows the system performance when both state transition and object emission vary. With greater variance, our model has significant im-

provement on both state and object estimation. We can conclude that if a workflow has larger variances on state transition distribution and object emission distribution, our system can achieve better performance.



(a) State Estimation (b) Object Estimation

Figure 5.7: Performance as both state transition and object emission distributions vary.

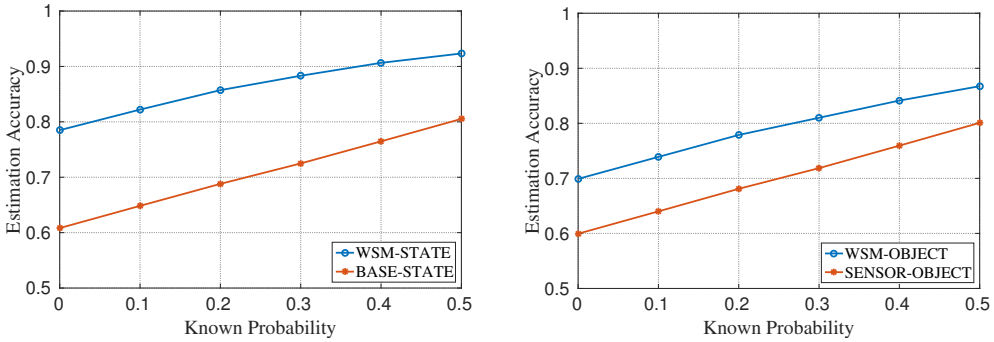


(a) State Estimation (b) Object Estimation

Figure 5.8: Performance as probability of known states varies

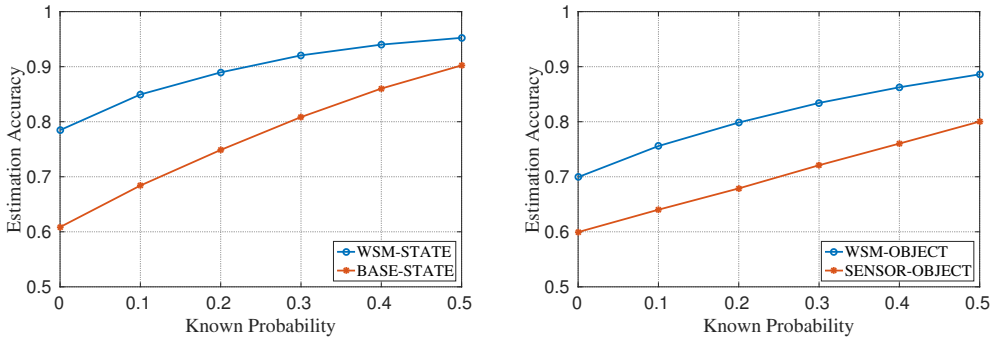
Finally, we evaluate our inference algorithm with partially known states and objects as supervision. In order to have a quantitative analysis, we use a parameter to control the probability that whether the state/object is known at time $t = 1...N$. In our experiments, the probability varies from 0 to 0.5, where 0 means no states or objects are known beforehand.

Fig. 5.8 shows the performance as probability of known states varies. With more percentage of states known, our model achieves better performance on state estimation, so does the baseline BASE-STATE. Still, the WSM model



(a) State Estimation (b) Object Estimation

Figure 5.9: Performance as probability of known objects varies



(a) State Estimation (b) Object Estimation

Figure 5.10: Performance as probability of known states and objects varies

consistently beats BASE-STATE. Objects estimation of WSM also improves because of better estimation of states.

Fig. 5.9 shows the performance as probability of known objects varies. With more percentage of objects known, our model has better performance on objects estimation, so does the baseline SENSOR-OBJECT, which takes account of the known objects as well. The WSM model still consistently performs better than SENSOR-OBJECT. States estimation of WSM also improves because of more accurate estimation of objects.

Fig. 5.10 shows the performance as probability of known states and objects varies. With higher known probability, our model has significant improvement on both state and object estimation. We can conclude that with more states and/or objects known as supervision, our model can achieve better performance.

CHAPTER 6

CASE STUDY EVALUATION

In this section, we apply our workflow-aware sensing model to develop a log service for human teams of first responders, called *emergency transcriber*. It constitutes an audio interface for reliably recording and disseminating situation progress as extracted from the teams' audio communications.

6.1 Experimental Settings

Workflow Information: We choose adult cardiac arrest [1] as our case of study. It strictly follows the emergency reaction algorithm shown in the Fig. 6.1, which includes a set of stages based on different conditions of the patients. In practical settings, when a patient is subject to cardiac arrest, multiple physicians and nurses operate around the patient at the same time, and medical orders are vocally communicated. The entire environment is noisy and chaotic. Commercial off-the-shelf speech recognition sensors often perform poorly in such environment.

System implementation: Our system consists of two major components. The first component is an existing speech recognizer sensor (ASR). Here we use Google Speech API [8]. It acts as an audio interface for carrying out the initial recognition of medical team's audio communications, as indicated by $R1$ in Fig. 6.2. Since the ASR does not have workflow information, it tries to use a general language model and an acoustic model to match the signal it hears, which leads to errors in recognition in noisy environments. $R1$ is then fed to our emergency transcriber, which consists of two modules; a *keyword matching* module and a *word recovery and state tracking* module.

The keyword matching module first applies keyword matching to match ASR output to the most similarly sounding keywords in our workflow. This is equivalent to find the keyword that has the maximum number of overlap-

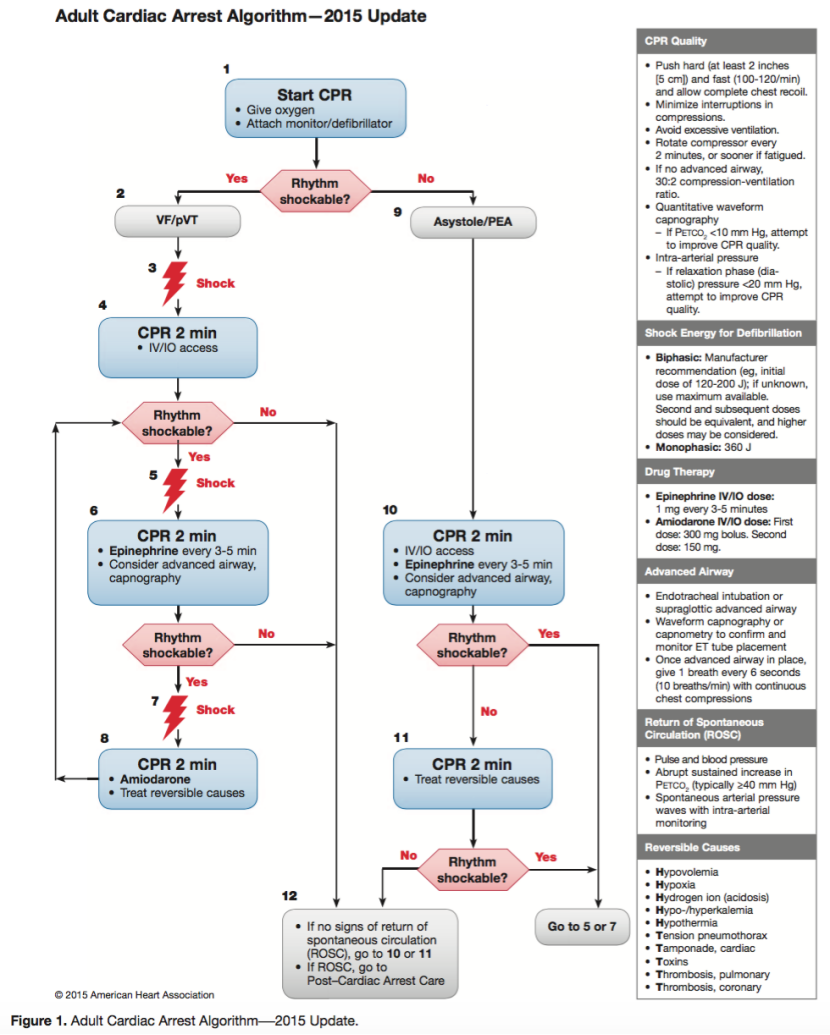


Figure 1. Adult Cardiac Arrest Algorithm—2015 Update.

Figure 6.1: Adult cardiac arrest workflow for resuscitation

ping phoneme characters with the sentence transcribed by the ASR. This is a *convolution* operation. Since both $R1$ and the keywords are in the form of English text, we convert $R1$ and all the keywords into their phoneme representations using a text synthesis software[9], and then calculate the convolution using Algorithm 3. The time complexity is $O(length(x) \times length(y))$.

Next, $R2$ is fed to the *word recovery and state tracking* module, where state-aware correction takes place, as described in Section 3. As an approximation, instead of training the ASR and getting it the classification confusion matrix (which is a very lengthy process), we apply the following equation to calculate each element in the confusion matrix:

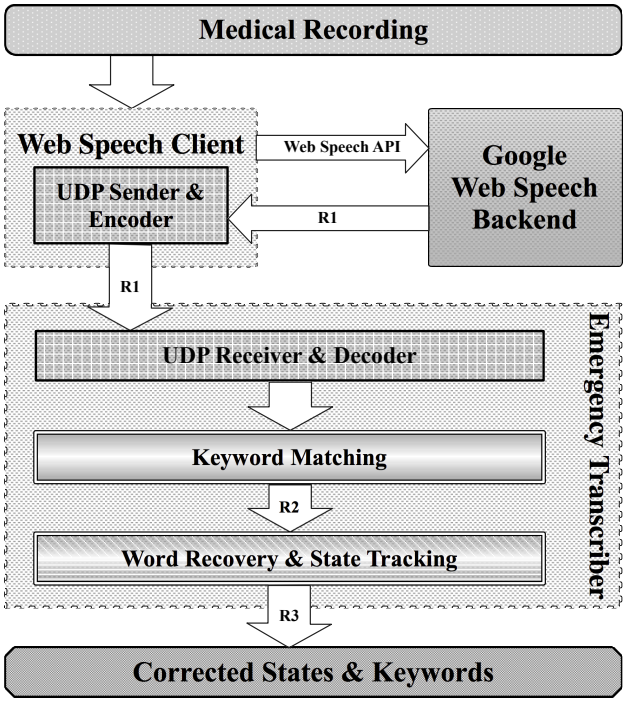


Figure 6.2: Architecture of emergency transcriber

$$sim(y_i|x_j) = (1 - \frac{LD_{i,j}}{\max(length(y_i), length(x_j)))})conv(y_i, x_j)$$

where $sim(y_i|x_j)$ represents the similarity between the observation word y_j and the true keyword x_i. Note that, $LD_{i,j}$ represents the Levensthein distance [10] between the phoneme representation of y_i and x_j. $length(y_i)$ and $length(x_j)$ represent their phoneme length, respectively. $conv(y_i, x_j)$ represents the convolution of their phoneme representations, which captures sub-phoneme overlapping between y_i and x_j. It is calculated according to algorithm 3. We then aim to recover the actual words spoken, and reveal the actual states traversed.

6.2 Experimental Results

We invited seven people (four are non-native English speakers) to record the script of a medical episode involving simulated emergency treatment of adult cardiac arrest that follows the workflow presented in Fig. 6.1. This script was designed by medical personnel from Carl Foundation Hospital in Urbana

Algorithm 3: Find maximum number of overlapping character between sentence **x** and keyword **y**

```
1:  maxlen ← 0
2:  for k = 0; k < length(x); k ← k + 1 do
3:      len ← 0
4:      i ← k
5:      j ← 0
6:      while i < length(x) and j < length(y) do
7:          if x[i] == y[j] then
8:              len ← len + 1
9:          end if
10:         i ← i + 1
11:         j ← j + 1
12:     end while
13:     if len > maxlen then
14:         maxlen ← len
15:     end if
16: end for
```

Illinois as part of a demonstration scenario of novel medical technologies. The script contained 21 sentences spoken during the simulated emergency. The script was re-enacted and the resulting audio was fed to the first component of our system.

A screenshot of our user interface is shown in Fig. 6.3. Notice that the leftmost text area shows the initial results $R1$ coming out of ASR. The dynamic graph in the middle tracks and visualizes the state-transition sequence in real time, with recovered keywords shown on the right side. The red circle indicates the current state and the blue circles indicate states that have been traversed in the workflow. We then added noise of different amplitudes to the original audio file, and sent it through the same pipeline. The result with average accuracy and standard deviation is shown in Fig. 6.4.

As can be seen from the result, when noise-free, the accuracy of the existing speech recognization sensor is 76.69%. Keyword matching increases this accuracy by comparing the output to the entire workflow vocabulary at any stage of the workflow. Moreover, with the workflow topology information accounted for, the word recognition accuracy increases to 95.24%, with 100% state recognition accuracy, which bolsters the claim that workflow knowledge can enhance sensing accuracy. When noise (Gaussian white noise) is added

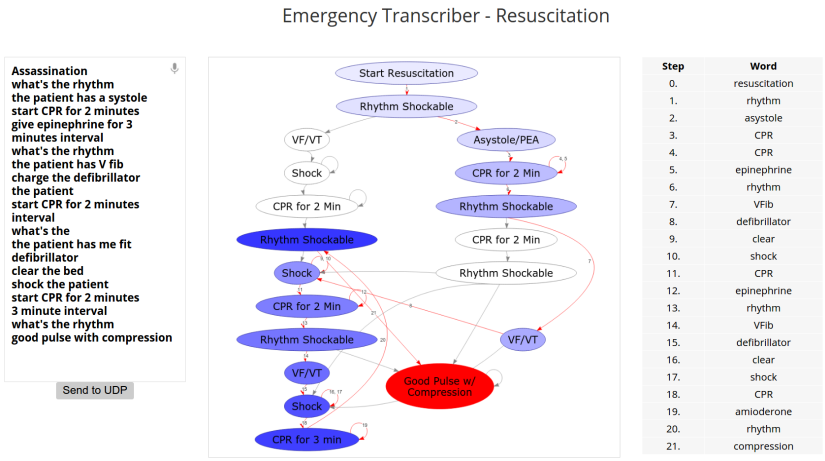


Figure 6.3: User interface of emergency transcriber

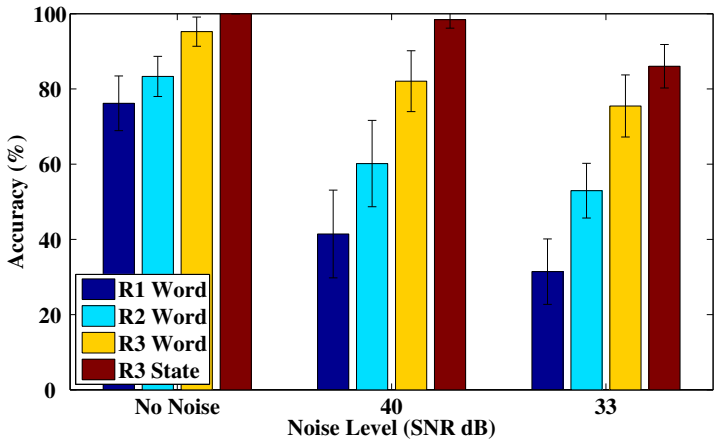


Figure 6.4: Recognition accuracy on different noise levels

to the original voice signals with a SNR (Signal-to-Noise Ratio) of $40dB$, the accuracy of the word recognition of $R1$, $R2$, and $R3$ decreases, but similar trends are observed. The situation is similar when the signal to noise ratio goes up to $33dB$. These results show that the emergency transcriber is a useful aid in recording emergency procedures in a range of noisy environments.

CHAPTER 7

RELATED WORK

Classification techniques on sensor data have been widely studied. For example, [11] studies data classification problem in wireless sensor networks. It proposed a classification approach in combining local classifier to form a global classifier to achieve high accuracy. [12, 13] proposed hierarchical aggregate classification methods to achieve high accuracy in lack of energy and label information. Our work differs from the exiting work in the sense that it takes the workflow information into consideration to enhance sensing accuracy with unreliable sensors and environmental noise. Besides it also keeps track of the states that have been traversed in the workflow.

Our workflow-aware sensing model is inspired by the Hidden Markov Model (HMM) [7], which is widely used in the area of speech recognition [14, 15, 16]. However, traditional HMM models the state transition between different phonemes as Hidden Markov Process. Our model is different because the observations acquired by the sensing system are not accurate. In order to take that into consideration, we combine the confusion matrix of the sensor with the HMM layer and find the optimal sensing object as well as the hidden states as a whole. For the case study specifically, the hidden states refer to the stages where physicians have been working on.

We apply our scheme in the area of speech recognition [17, 18] under medical environment. There are several commercialized speech recognition software available for clinical documentation, such as [19, 20]. Our approach is complementary to the above-mentioned automatic speech recognizers (ASRs) because it considers external workflow constraints when doing speech recognition, and the workflow information is free from sensor errors and environment noise. It can act as a light-weight wrapper outside the ASRs for any specific use case, thus our scheme has the advantage of good compatibility and portability.

Compared to our previous work [6], this thesis proposes a new inference

algorithm (Algorithm 2) to handle cases with partially known states and/or objects as supervision. We also describe how to learn and smooth model parameters. In simulations, we rewrite the simulator in C++ and conduct experiments in larger scales (100,000 random workflows). Instead of dividing workflows into graph and tree, we simply adopt randomly generated graph (tree can be seen as special graph) as workflow. We add a baseline for state reference and show state and object estimation results in separate figures. We add a simulation experiment on performance of model with variable state transition and object emission probability matrix. We also evaluate the new proposed inference algorithm (Algorithm 2) by varying the probability of known states and/or objects.

CHAPTER 8

CONCLUSION

In this thesis, we describe a general methodology for enhancing sensing accuracy in cyber-physical systems that involve human domain workflows in a noisy physical environment. We propose a workflow-aware sensing model which can jointly infer the optimal sensing measurements and state transition sequence by exploiting human workflow information. Simulation results show that our model outperforms the accuracy of commercial off-the-shelf sensors. We instantiate our idea by conducting a case study in medical emergency environment and demonstrate that our model can improve speech recognition and state tracking.

REFERENCES

[1] M. S. Link, L. C. Berkow, P. J. Kudenchuk, H. R. Halperin, E. P. Hess, V. K. Moitra, R. W. Neumar, B. J. O'Neil, J. H. Paxton, S. M. Silvers et al., "Part 7: Adult advanced cardiovascular life support 2015 american heart association guidelines update for cardiopulmonary resuscitation and emergency cardiovascular care," *Circulation*, vol. 132, no. 18 suppl 2, pp. S444–S464, 2015.

[2] M. Kastner, M. W. Saleh, S. Wagner, M. Affenzeller, and W. Jacak, "Heuristic methods for searching and clustering hierarchical workflows," in *Computer Aided Systems Theory-EUROCAST 2009*. Springer, 2009, pp. 737–744.

[3] A. Avanes and J.-C. Freytag, "Adaptive workflow scheduling under resource allocation constraints and network dynamics," *Proceedings of the VLDB Endowment*, vol. 1, no. 2, pp. 1631–1637, 2008.

[4] C. Talcott, "Cyber-physical systems and events," in *Software-Intensive Systems and New Computing Paradigms*. Springer, 2008, pp. 101–115.

[5] I. Lee and O. Sokolsky, "Medical cyber physical systems," in *Proceedings of the 47th Design Automation Conference*. ACM, 2010, pp. 743–748.

[6] Y. Gao, S. Hu, R. Mancuso, H. Wang, M. Kim, P. Wu, L. Su, L. Sha, and T. Abdelzaher, "Exploiting structured human interactions to enhance estimation accuracy in cyber-physical systems," in *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM, 2015, pp. 60–69.

[7] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[8] Google, "Google speech api," 2016. [Online]. Available: https://www.google.com/intl/en/chrome/demos/speech.html

[9] eSpeak, "Speech synthesizer," 2016. [Online]. Available: http://espeak.sourceforge.net/

[10] G. Navarro, "A guided tour to approximate string matching," *ACM computing surveys (CSUR)*, vol. 33, no. 1, pp. 31–88, 2001.

[11] X. Cheng, J. Xu, J. Pei, and J. Liu, "Hierarchical distributed data classification in wireless sensor networks," *Computer Communications*, vol. 33, no. 12, pp. 1404–1413, 2010.

[12] L. Su, J. Gao, Y. Yang, T. F. Abdelzaher, B. Ding, and J. Han, "Hierarchical aggregate classification with limited supervision for data reduction in wireless sensor networks," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2011, pp. 40–53.

[13] L. Su, S. Hu, S. Li, F. Liang, J. Gao, T. F. Abdelzaher, and J. Han, "Quality of information based data selection and transmission in wireless sensor networks," in *Real-Time Systems Symposium (RTSS), 2012 IEEE 33rd*. IEEE, 2012, pp. 327–338.

[14] M. Gales and S. Young, "The application of hidden markov models in speech recognition," *Foundations and trends in signal processing*, vol. 1, no. 3, pp. 195–304, 2008.

[15] B. H. Juang and L. R. Rabiner, "Hidden markov models for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.

[16] S. Young, "Hmms and related speech recognition technologies," in *Springer Handbook of Speech Processing*. Springer, 2008, pp. 539–558.

[17] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech communication*, vol. 34, no. 3, pp. 267–285, 2001.

[18] R. P. Lippmann, "Speech recognition by machines and humans," *Speech communication*, vol. 22, no. 1, pp. 1–15, 1997.

[19] Nuance, "Dragon medical speech recognition," 2016. [Online]. Available: http://www.nuance.com/for-healthcare/dragon-medical/index.htm

[20] EvolveMed, "Talkchart," 2016. [Online]. Available: https://www.talkchart.com/